IN THE SPECIFICATION:

Please amend paragraphs **[0026]**, **[0042]**, **[0044]**, **[0053]**, and **[0065]** as follows:

[0026] The network 100 preferably includes multiple dispersed datacenters (DC1-DC4). Some of the datacenters may be located, geographically, close to each other, and others may be located far from the other datacenters. Furthermore, one of the datacenters may be the primary source of new files to be distributed to the other datacenters, or alternately, the generation of new files may be shared by two or more of the datacenters. Each datacenter (DC1-DC4) preferably includes multiple racks. For example, datacenter one (DC1) includes multiple racks (Rack 1 – N). Physically, the racks may include frames or cabinets into which the slaves and other components, such as switches, are mounted. Each rack preferably includes multiple slaves. For example, the first rack (Rack 1) includes multiple slaves (Slave 1-16), the second rack (Rack 2) includes multiple slaves (Slave 17-31, and the third rack (Rack 3) includes multiple slaves (Slave 32-~~N~~ M). The slaves can include any electronic or electromechanical unit of equipment, but in a preferred embodiment are computers. In other embodiments the slaves may include computer controlled devices, such as network attached storage devices.

[0042] The data files 228 distributed include very large files. To make the transfer of the large files more manageable, and to avoid having to retransmit an entire file when a failure occurs midway through a file transfer, the data files are preferably segmented into blocks, as shown in Figure 2D. In one embodiment each block has a size of 16MB (Megabytes) or less. More specifically, large files are both generated and copied in blocks of 16MB, until the last block of the file has been received, which has whatever size (of 16MB or less) required to complete the file copy process. It should, however, be appreciated that the block sizes may be any suitable size and/or of different sizes, as determined in accordance with predefined criteria.

[0044] Returning to Figure 2A, the verification procedures ~~228~~ 240 are used to verify the integrity of the data received by, or stored on, the slaves. In one embodiment, the verification procedures 228 include a procedure for comparing each received block with a received checksum file 230. In a preferred embodiment, each data file 228 has a

corresponding checksum file 230, as shown in Figure 2~~C~~ D. The checksum file 230 includes a checksum value for each block of the data file. In a preferred embodiment, each checksum is a cumulative checksum that is a function of (1) the data in the data file block corresponding to the checksum, and (2) the cumulative checksum for all the previous blocks of the file, if any. The checksum value may be a cyclic redundancy check (CRC) checksum, generated using a predefined polynomial function (e.g., the well-known 32-bit Ethernet CRC polynomial) or any other suitable checksum function.

[0053] The memory 362 also preferably includes a system hierarchy table 380, a system resources table 382, a pending transmissions table 384, a state table 390, failure determination procedures 392, resource allocation and scheduling procedures 394, and a cache 396. The system hierarchy table 380 is used by the master to determine the proximity of slaves to one another. The system resources table 382 is used by the master to determine the bandwidth resources for each communication path in the switched network, and the resources currently in use or reserved for use. The pending transmissions table 384 is used to keep track of all current or pending transmissions. The state table 390 is used by the master to determine which system resources have failed, as well as which files or blocks of files have been received by each slave and ~~while~~ which files or file blocks are still needed by each slave. The failure determination procedures 392 are used by the master for determining whether a resource has failed. The resource allocation and scheduling procedures 394 are used to schedule data transfers between slaves, as described in further detail below in relation to Figures 4 and 5A-5D. The cache 396 is used for temporarily storing data.

[0065] In addition, each resource has an associated transmission attempts count (Attempts), as well as a failure count (Failure Count) that can be incremented or decremented by the failure determination procedures 392 (Figure 3A). In use, the failure determination procedures 392 (Figure 3A) are used to determine if a particular resource is likely to have failed. Each time a copy operation is attempted, the corresponding Attempts Count for each device or resource involved in the data transfer is incremented. Each time a copy operation fails, the Failure Count is incremented for each resource that is potentially responsible for the failure. In some embodiments, the master receives an indication from the receiving slave indicating which resource(s) are believed to be responsible for the failure. The failure determination procedures 392 (Figure 3A) then increment the failure count (Failure Count)